

Track Allocation in Freight-Train Classification with Mixed Tracks*

Markus Bohlin¹, Holger Flier², Jens Maue², and Matúš Mihalák²

¹ Swedish Institute of Computer Science, Kista, Sweden

markus.bohlin@sics.se

² ETH Zürich, Institute of Theoretical Computer Science, Switzerland

{firstname.lastname}@inf.ethz.ch

Abstract

We consider the process of forming outbound trains from cars of inbound trains at rail-freight hump yards. Given the arrival and departure times as well as the composition of the trains, we study the problem of allocating classification tracks to outbound trains such that every outbound train can be built on a separate classification track. We observe that the core problem can be formulated as a special list coloring problem in interval graphs, which is known to be NP-complete. We focus on an extension where individual cars of different trains can temporarily be stored on a special subset of the tracks. This problem induces several new variants of the list-coloring problem, in which the given intervals can be shortened by cutting off a prefix of the interval. We show that in case of uniform and sufficient track lengths, the corresponding coloring problem can be solved in polynomial time, if the goal is to minimize the total cost associated with cutting off prefixes of the intervals. Based on these results, we devise two heuristics as well as an integer program to tackle the problem. As a case study, we consider a real-world problem instance from the Hallsberg Rangerbangård hump yard in Sweden. Planning over horizons of seven days, we obtain feasible solutions from the integer program in all scenarios, and from the heuristics in most scenarios.

1998 ACM Subject Classification G.1.6 Optimization; I.2.8 Problem Solving, Control Methods, and Search

Keywords and phrases Algorithms, Complexity, Graph Theory, Railways, Scheduling, Shunting, Train Classification, Train Marshalling, Transportation

Digital Object Identifier 10.4230/OASICS.ATMOS.2011.38

1 Introduction

A rail-freight transportation network is used to transport goods between two points in the network. Described in a very simplified way, such a network consists of terminals connected with tracks. Besides freight trains that consist only of cars going from one origin to one destination (e.g., trains transporting coal or ore), most freight trains consist of cars with various origins and destinations. In order to transport each car to its destination, the freight trains are disassembled into individual cars from which new outbound trains are formed. This process is called *classification* (also *marshalling* or *shunting*), and is conducted in so called *classification yards* (also known in the literature as *marshalling* or *shunting yards*), which are intermediate stops (terminals) in the network.

* This work was supported by Trafikverket under FUD grant F 09-11546/AL50 and by the Swiss National Science Foundation (SNF) under grant 200021-125033/1.



© Markus Bohli, Holger Flier, Jens Maue and Matúš Mihalák;
licensed under Creative Commons License NC-ND

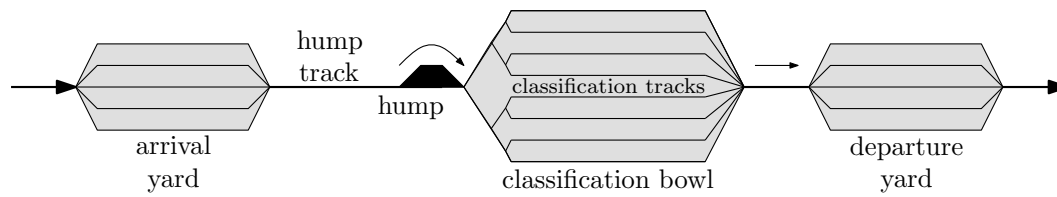
11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems.

Editors: Alberto Caprara & Spyros Kontogiannis; pp. 38–51

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** A schematic layout of a typical hump yard.

Train classification is a crucial element in the overall goal of an efficient rail-freight transportation as this process is often the bottleneck of the freight transportation and naturally limits the throughput in the network. In this paper, we study one specific process of forming outbound trains from the cars of inbound trains within a classification yard.

A classification yard typically consists of an *arrival yard*, a *hump track* with a *hump*, a *classification bowl*, and a *departure yard*. Therefore, classification yards with a hump are often called *hump yards*. A typical layout of a hump yard is illustrated in Figure 1. The *arrival yard* is a composition of tracks used for storing cars of inbound trains, where all cars of an inbound train are decoupled and stored on a single track. Similarly, a *departure yard* is a composition of tracks for storing outbound trains. The outbound trains are formed in the *classification bowl*, which consists of *classification tracks*. The hump is connected to the classification tracks by a set of switches such that a car or an engine can reach any classification track from the hump. To form the desired composition of cars on the classification tracks, two operations are used: the *pull-out* and the *roll-in* operation. In the pull-out operation, the cars of a specified classification track are coupled and pulled by the engine over the hump onto the hump track; there, the cars are decoupled and are ready for the subsequent roll-in. During the roll-in operation, the (decoupled) cars on the hump track are pushed over the hump and roll into the classification bowl by means of gravity only; each individual car is guided to a desired classification track by appropriately setting the switches of the classification bowl. In a hump yard, a pull-out operation is always followed by a roll-in operation. When the formation of an outbound train is finished, its cars are coupled and moved by an engine to the departure yard.

Given the arrival and departure times of the inbound and outbound trains, respectively, as well as the cars belonging to these trains, the *operational plan* of a hump yard needs to decide the movements of every car in the hump yard by the means of roll-ins and pull-outs in order to achieve the desired formation of every outbound train before its departure time. There may be various constraints on the formation of the outbound trains. In particular, there are situations when the order of cars within a train is important. In this paper we only consider the case in which no particular order of cars within an outbound train is required. This is usually the case for freight trains which are not delivering goods to the final destination but to another classification yard. A general operational plan would also need to decide the time points when cars enter (from the arrival yard) and leave (to the departure yard) the classification bowl. In this paper we consider a less ambitious goal, and only consider the case when these time points have been decided. We thereby focus solely on planning the movements of the cars within the classification bowl.

In this paper, we restrict ourselves to a particular class of operational plans which is a generalization of current customs that are used in several hump yards in Europe. In these operational plans, most classification tracks are reserved for at most one outbound train at any point in time. Furthermore, a few classification tracks, called *mixed tracks*, are used to store a mix of cars (which may arrive long before their planned departure) of different

outbound trains in order to increase the capacity of the classification bowl. These mixed tracks are pulled-out, e.g., at some fixed time-points per day, in order to distribute cars onto classification tracks. A car can only go to its track once it is reserved for its outbound train. The remaining cars go back to the mixed tracks. At present, operational plans are hand-made. In order to compute such plans automatically, we present several results in this paper, including our modeling approach, relation to algorithmic theory, and initial computational results suggesting that operational plans can be found in less than 20 minutes.

The paper is structured as follows. We first define the mixing problem formally in Section 1.1, and then review the related work and the best practice in Section 1.2. We connect the mixing problem to various coloring problems of intervals in Section 2, where we also develop two heuristics for the problem. We then present our mixed-integer program for the mixing problem in Section 3. The experimental results are discussed in Section 4.

1.1 Problem definition

We consider a classification bowl consisting of k classification tracks $\theta_1, \dots, \theta_k$, where a classification track θ has length ℓ_θ . Furthermore, some of the tracks of the classification bowl are used for mixing cars of different outbound trains. We denote the sum of their lengths as ℓ^{mix} and refer to them simply as the mixed track. There are n^{in} inbound trains, which are to be formed into n outbound trains, where each car of an inbound train belongs to exactly one outbound train. Individual cars having the same inbound and outbound trains are handled as a single unit, a car *group* g , which has physical length ℓ_g . Hence, an outbound train j consisting of a set \mathcal{G}_j of car groups has length $\sum_{g \in \mathcal{G}_j} \ell_g$. For each outbound train j we need to *assign* a classification track θ to j on which it will be formed. The track θ has to have sufficient length, i.e., the length ℓ_θ must be at least the length of the outbound train. Every outbound train j has a fixed time o_j when it leaves the classification bowl (to the departure yard). Thus, by this time, all cars of the train j have to be on its assigned track. For each outbound train j we need to decide a time interval (s_j, o_j) during which the respective assigned track θ is reserved solely for cars of train j . Every inbound train i has a roll-in time r_i at which the cars of the inbound train are rolled in (from the arrival yard) over the hump (into the classification bowl). Each car group g is rolled-in either to the mixed track, or to the classification track θ assigned to the outbound train j , to which group g belongs: if at time r_i the assigned track θ is already reserved (i.e., if $r_i > s_j$) then group g is rolled-in to the classification track θ , otherwise it is rolled-in to the mixed track. A car that is rolled-in to the mixed track needs to be pulled-back from the mixed track over the hump and rolled-in to the assigned track during the time interval that is reserved for its outbound train on that track. For this purpose, the mixed track is pulled out at fixed times p_1, \dots, p_m . At such a time, all cars of the mixed track are subsequently rolled-in either to a classification track (if the respective assigned track has been already reserved), or back to the mixed track (if the respective track has not yet been reserved). We call each time interval between two consecutive pull-outs a *period*.

In this paper we consider the problem of assigning a classification track θ to every outbound train j , as well as deciding the time s_j when the assigned classification track should be reserved for train j , such that all outbound trains can be formed. We will refer to this problem as the *mixing problem*. Observe that the schedule of the hump, which specifies the times of the roll-in and pull-out operations as well as the time o_j of each outbound train j , is fixed. Thus, the set of cars which are stored on the mixed track is determined by the choice of s_j for each outbound train j .

1.2 Related work

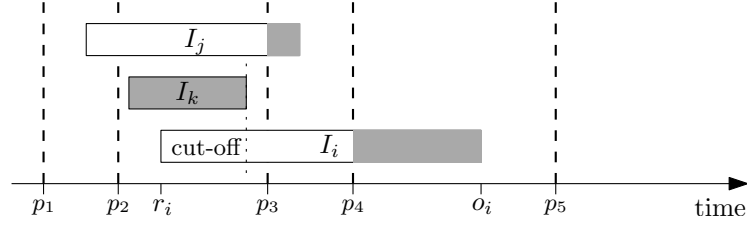
The particular problem that we consider in this paper has, as far as we know, not been studied before. There are various papers related to the problem of shunting both freight and passenger trains, but the solutions techniques are not applicable to the shunting problem when mixing is taken into account.

Many research efforts related to the operation of classification yards have been put in *sorting schemes* for sorting cars inside a classification bowl. Given a sequence of n cars labeled from 1 to n , the general goal of a sorting scheme is to form, by roll-in and pull-out operations, a sorted sequence of cars. Early literature considers sorting schemes that essentially perform the same sorting steps for any input sequence of a given length [11]. More recently, it has been studied how to utilize the “pre-sortedness” of the input in order to minimize the number of pull-out operations [6, 10], as well as variants thereof [5]. A recent survey by Gatto et al. [8] gives an overview of this topic.

A related problem is the parking of trams in the evening on tracks in depots such that the trams can leave the depot in the morning without any shunting operation [2, 7, 12]. Another related problem is the train scheduling at yards, i.e., the problem of assigning trains and train times for a set of rail lines and station stops. This problem was considered by He et al. [9], together with some operational planning at classification yards in China, although under different considerations than in our case.

In the problem considered in this paper, we have to decide for each outbound train both the classification track on and the time at which it will be formed. A related problem for passenger trains has been considered [4], in which more than one inbound train can be assigned to a track of a train station. The problem asks for an assignment of tracks to inbound trains such that the trains do not block each other when departing the train station.

Best Practice Today, the planning of hump yard operations is to a large extent done manually. In the Hallsberg yard in Sweden, where the data for our experimental evaluation was collected, detailed planning is done by the hump-yard staff one day at a time, usually during the morning when fewer trains arrive than in the afternoon. The allocation of tracks at the arrival yard and departure yard is performed manually, independently from other operations, and in advance by traffic-planning personnel, who are not directly involved with hump yard operation and detailed planning. However, frequent communication between the different groups happens, since the allocation of arrival and departure yards and the yard operation planning is interdependent and cannot be done in full isolation. The typical practice at Hallsberg is to use the same roll-in order as the arrival time order and the same roll-out order (onto the departure yard) as the departure time order. In this paper we follow this practice in that we assume the roll-in times and roll-out times (into and out of the classification bowl, respectively) to be given as part of the input. For the experimental evaluation where we do not have this data, we compute these times as described in Section 4.2. Also, a common practice is to pull-out all mixed tracks together, although it could be beneficial to pull-out the mixed tracks independently. For our purposes, we treat these mixed tracks as one virtual mixed track, where we set the length of the track and the duration of a pull-out operation accordingly.



■ **Figure 2** Three outbound trains i , j and k induce the (mutually intersecting) intervals I_i , I_j and I_k , here depicted as rectangles. The pull-out of the mixed tracks happens at times p_1, \dots, p_5 . The uncuttable part of an interval is defined by the last pull-out time of the mixed track that is contained in the interval (depicted in gray). If we assume two available tracks, then the three intervals cannot be assigned to the tracks without cutting off. Here, a cut-off of the interval I_i at the end point of I_k allows a placement of I_i' and I_k on the same track. For simplicity, technical setup times have been omitted in this figure.

2 Relation to interval-coloring problems

The mixing problem can be seen as a family of specific coloring problems of intervals. In this section, we give complexity results based on these relations and devise two heuristics for our problem that we experimentally evaluate on real-world data in Section 4.

Recall that in the mixing problem we are asked to determine for each outbound train i a track θ and a time interval I_i' , during which the track is reserved exclusively for the formation of that train. Observing the roll-in times of the inbound trains, we can obtain for every outbound train i a time interval $I_i = (arr, dep)$ in which cars of the outbound train arrive to the classification bowl, i.e., $arr = \min_{g \in \mathcal{G}_i} r_g$ (where r_g is the roll-in time of the inbound train to which car group g belongs, and \mathcal{G}_i is the set of car groups of train i), and $dep = o_i$ is the time when the train i leaves from its classification track to the departure track. Thus, without loss of generality, the time interval I_i' is a sub-interval of I_i of the form $I_i' = (arr', dep)$, $arr' \geq arr$, i.e., we *cut-off a prefix* of I_i to obtain I_i' . We cannot cut-off an arbitrary prefix: whenever $arr' \neq arr$, there has to be a pull-out of the mixed track between arr' and dep , because we require that every car that is sent to the mixed track is at some point brought to the actual track θ (before the train departs). This, together with some technical setup times which we do not describe here for simplicity, induces for every interval I_i an *uncuttable part* of I_i , i.e., a suffix of I_i during which all not yet rolled-in cars of the outbound train have to be rolled in directly onto its classification track. The requirement that every track θ is at any time reserved for at most one train translates into the condition that, whenever trains i and j are assigned the same track θ , the corresponding intervals I_i' , I_j' do not overlap. Here and in the following, two intervals overlap if they intersect in more than one point. Figure 2 illustrates our discussion.

Our problem thus translates to the problem of assigning a track θ to every outbound train i and cutting off a prefix of every interval I_i to obtain a cut-off interval I_i' such that no two cut-off intervals of two trains assigned to the same track overlap. Assuming the cutting-off of intervals has been made, the problem of assigning tracks of sufficient length to the outbound trains can be seen as a list-coloring problem of the intervals: each train i has a list L_i of classification tracks to which it fits, each track represents a color, and we are asked to color every interval I_i with a color from the list L_i such that any two overlapping intervals I_i' , I_j' need to be assigned different colors.

In general, the list-coloring problem of intervals is NP-complete. In our case, the lists do not have arbitrary structure: Assume w.l.o.g. that the classification tracks $\theta_1, \dots, \theta_k$ are

ordered increasingly by length. For each outbound train i , let $\mu(i)$ indicate the smallest track on which it fits (we assume that every train fits on θ_k); the list L_i is then just $\{\theta_{\mu(i)}, \dots, \theta_k\}$. Such a list-coloring problem is called a μ -coloring problem. It is known that the μ -coloring problem for interval graphs is NP-complete, see [3]. As a consequence we immediately obtain the following theorem:

► **Theorem 1.** *Finding a feasible track allocation for the mixing-problem is NP-complete even for instances where the capacity of the mixed track ℓ^{mix} is zero, or where ℓ^{mix} is unlimited and all intervals may have arbitrary uncuttable parts.*

Proof. Observe first that if there is no capacity on mixed tracks then no car can be sent to a mixed track and thus no interval can be cut-off. If on the contrary ℓ^{mix} is unlimited, we may, without loss of generality, assume that every interval I_i has been cut-off in a maximal possible way and I'_i is thus the uncuttable part of I_i . It is now easy to see by the above discussion how to transform any instance of the μ -coloring problem for intervals to a corresponding instance of the mixing-problem. ◀

Despite its NP-complete core, the practical complexity of the mixing-problem strongly depends on the distribution of the length of both the classification tracks and the outbound trains. If, for example, each train fits on each track, and the capacity of the mixed tracks is zero, then our problem reduces to the problem of coloring an interval graph, which is well-known to be polynomially solvable by a simple greedy algorithm. The heuristic in the following section is based on this observation.

If we assume that each train fits on each track and sufficient length of the mixed track, a natural heuristic for the mixing problem would be to minimize the total number of cars that are sent to the mixed tracks and that allows a feasible track allocation. In the following we show that this problem can be solved in polynomial time. Further below, we will use this result to devise an improvement heuristic for the mixing problem.

► **Theorem 2.** *In case of uniform and sufficient track lengths, the problem of finding a feasible track allocation that minimizes the sum of all cars sent to the mixed track over all time periods is solvable in polynomial time.*

Proof. Assume there are k tracks and n outbound trains, each with a time interval as described above. Observe that if the trains are assigned to tracks then computing the minimum number of cars that need to be sent to the mixed tracks in order to make this assignment feasible is a trivial task. To see this, consider a classification track θ and train i with interval $I_i = (arr_i, dep_i)$ such that the train is last to leave the track, i.e., for any other train i' with time interval $(arr_{i'}, dep_{i'})$ assigned to track θ we have $dep_{i'} < dep_i$. The minimum number of cars of train i that we need to send to the mixed track are the cars that arrive in time period $(arr_i, dep_{i'})$ where $dep_{i'}$ is the departure time of train i' that departs from the classification track second to last, i.e., just before i . We can proceed similarly with the cars of train i' by sending to the mixed track all cars of train i' that arrive in time interval $(arr_{i'}, dep_{i''})$, where i'' is the train leaving the classification track just before i' . We can proceed recursively to determine the minimum number of cars that need to be sent to the classification tracks. Therefore, we can see our problem as finding for every train i its direct predecessor i' on its assigned classification track. The actual assignment of a track to trains is done by introducing a *phantom* train i_θ for every track θ . Thus, if train i is assigned a phantom train i_θ as the direct predecessor of i , then we interpret this as assigning train i to the classification track θ . In this modified setting where every train is asked to have a predecessor (a real train or a phantom train), our problem can be reduced to an assignment

problem, i.e., to finding a minimum-weight matching in the following complete bipartite graph: the (real) trains form one part of the bipartition, and the real trains together with the phantom trains form the other part of the bipartition; the weight of the edge connecting train i from the first part with (phantom) train i' is the minimum number of cars that need to be sent from i to the mixed tracks in order to allow train i' to be an immediate predecessor of train i on a classification track (or the weight is ∞ , if i' cannot be a predecessor of i). ◀

2.1 A clique-based heuristic

We present a heuristic for the mixing problem in which we will iteratively decide a coloring of the intervals, and cutting off the (problematic) intervals. The guiding goal will be to keep the total length of all cars sent to the mixed track, summed over all periods, low. Ideally, the heuristic would find a solution to the mixing problem that is feasible w.r.t. the capacity of the mixed track in each period. Recall that for every outbound train i , we need to choose a suffix I'_i for each interval I_i and color it with a color from the list $L_i = \{\theta_i \mid i \geq \mu(i)\}$. We assume, again, that every outbound train fits on the largest track θ_k .

Coloring Intervals We first color the intervals in a greedy way, mimicking the greedy coloring of interval graphs without lists. We assume that we have infinitely many colors available (i.e., not only k). We start by sorting the intervals in a non-decreasing order of their starting time point. We color the intervals in this order and assign each interval I_i the smallest *non-conflicting color* that is at least $\mu(i)$. At this point, a non-conflicting color is a color such that no interval that has been colored and overlaps with I_i is colored with it.

In this way, we guarantee that every interval I_i is assigned a color at least $\mu(i)$. If we do not use more colors than k , we have found a list-coloring and can stop the iterations. Otherwise, the coloring uses more colors than k , and we proceed with the cutting off.

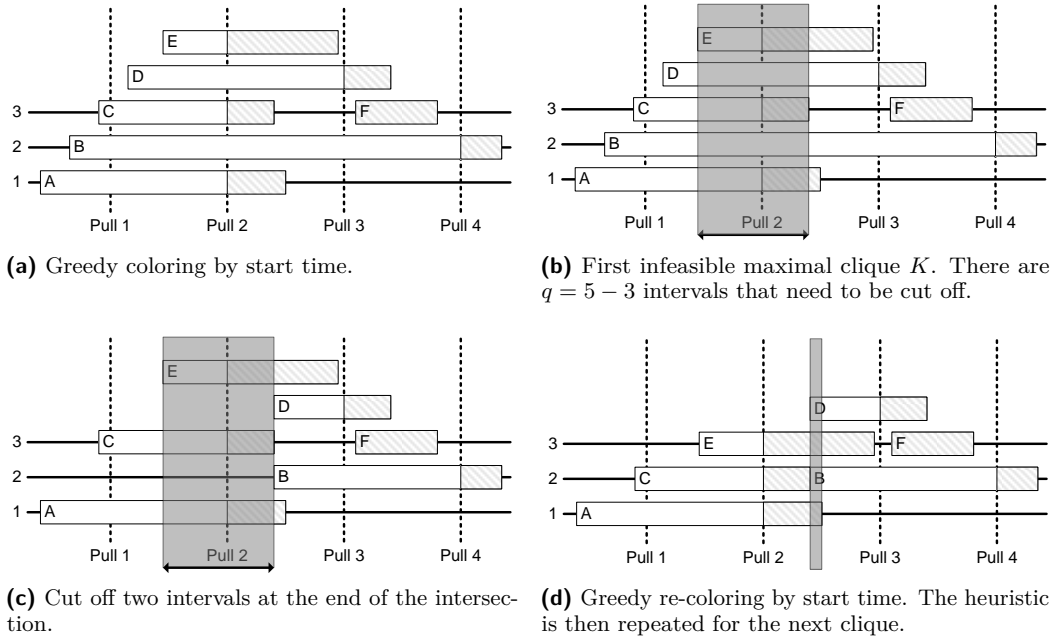
Cutting off Intervals If there is an interval I_i that is colored with a color $c > k$, then I_i overlaps with intervals that are together assigned every color in $\mu(i), \mu(i) + 1, \dots, k, \dots, c$. These intervals mutually intersect and thus form a clique K . We find a maximal clique containing an interval of the largest assigned color c . Let $q := c - k$, i.e., the number of intervals of K that use a color $c > k$. The heuristic tries to reduce the size of the clique by cutting off q intervals of K , in order to be able to increase the set of available colors for each of those intervals.

The cut-offs are computed as follows. First, the intersection of all intervals in K is computed, which is an interval by itself. Let t denote the end of this interval. If possible, cut off q of the intervals of K at point t . In particular, cut those q intervals of K that minimize the resulting additional usage of the mixed tracks. We iterate the procedure (coloring and cutting off) with the newly cut-off intervals.

This heuristic is illustrated in Figure 3. Note that the heuristic does not guarantee to find an allocation that is feasible with regard to the capacity of the mixed tracks.

2.2 An improvement heuristic

Once a feasible assignment of tracks to outbound trains exists, one can furthermore try to improve the solution towards a local optimum. In particular, we are interested in minimizing the total number of cars (over all periods) sent to the mixed track, which we call the *extra roll-ins* for short, as will be motivated in Section 3.2.



■ **Figure 3** First steps of the interval coloring heuristic involving trains A-F, and three classification tracks. Gray areas of the intervals indicate allocations which cannot be cut-off, as there is no pull-out (dotted vertical lines) of the mixed track. The dark grey rectangle depicts the intersection of the respective clique of intervals.

The heuristic is based on two observations. First, when looking at a feasible solution, one observes that both the tracks and the trains can be partitioned into subsets, called *buckets*, such that each train in a bucket fits on all tracks in the same bucket. Secondly, given a feasible solution, one can minimize the mixed-track usage for each bucket of tracks independently. Because within a bucket, all assigned trains fit on all tracks, it suffices to solve an assignment problem for each bucket, as detailed in Theorem 2, in order to find an optimal reassignment of trains to tracks within that bucket. Note that also this heuristic does not guarantee feasibility regarding the capacity of the mixed tracks.

3 An integer program for the mixing problem

In order to compute exact solutions for the mixing problem, we design an integer program. Recall that the mixing problem asks to find an assignment of long-enough classification tracks to the outbound trains, and for each outbound train a (conflict-free) time reservation of its assigned track, such that all outbound trains can be formed on time and the capacity of the mixed track is not exceeded.

In the following model, each train i is associated with binary variables x_{is} for each possible starting time point s of the reservation of its assigned classification track, and binary variables $y_{i\theta}$ for each possible classification track θ to which it may be assigned.

3.1 Capacity of the mixed tracks

To limit the amount of used mixing capacity, we first note that the set of mixed cars can only change when a car group is rolled in to the classification bowl, or at a pull-out. In addition, a car group rolled in to a mixed track at time s will stay mixed at least until the first pull-out p_s^+ that is scheduled after s . Therefore, it suffices to ensure feasibility of the mixed track usage at the end of each period, when the maximum usage within that period is attained. Let $\mathcal{X}_{is} = \{g \mid g \in \mathcal{G}_i, r_g < s\}$ be the set of car groups of an outbound train i that are mixed as a result of i starting at time s . We now need to consider which group of cars in train i are mixed at a certain pull-out p , given a start time s for i . To determine this we will check whether the prefix of I_i that is cut-off contains p , and which groups from \mathcal{X}_{is} are rolled in before p .

Let \mathcal{P}_i be the set of *valid pulls* for train i , which occur after the first group roll-in and before the time the train needs to start preparations for departure. Furthermore, let \mathcal{S}_i be the set of *valid start-times* for train i , which is the union of \mathcal{P}_i and the set of the roll-in times $\{r_g \mid g \in \mathcal{G}_i\}$. Now assume we have a train i with a valid start time s and a pull $p \leq p_s^+$. If i starts at s , then all groups $g \in \mathcal{X}_{is}$ (which have to be mixed when i starts at s) stay on the mixed track until the pull-out p_s^+ . Therefore, a group g will be mixed during the period ending at p if $r_g < p$.

Formally, we let $\mathcal{A}_p = \{(i, s) \mid 1 \leq i \leq n, p \in \mathcal{P}_i, s \in \mathcal{S}_i, p \leq p_s^+\}$ be the set of pairs of all trains and start-times possibly affecting a pull-out p , where n is the number of outbound trains. Given \mathcal{A}_p we can now define the mixed capacity constraints as shown below in Equation (5). Informally, this equation states that for all trains i and start times s affecting a pull-out p it holds that if the reservation of the classification track of i starts at s then the total length of the groups that arrive before p may not exceed the mixing capacity ℓ^{mix} .

3.2 Counting extra roll-ins

For the purpose of our research with the Swedish traffic administration authority Trafikverket, the goal of yard operation planning was to minimize unnecessary labor and infrastructure wear. The current practice at the yard reflects a policy where the goal is to roll in cars as soon as possible, and where cars are mixed if they arrive “early” compared to their planned departure. In practice, this policy leads to many cars being unnecessarily mixed and subsequently pulled-out.

As the objective of the optimization problem, we chose to minimize the number of extra roll-ins needed for yard operation, which corresponds to the number of times a car is sent to the mixed track. This number, c_{is} , can be easily calculated for each train i and each possible starting time point s as $c_{is} = \sum_{g \in \mathcal{X}_{is}} n_g |\mathcal{P}_{isg}|$, where \mathcal{X}_{is} is defined as above and $\mathcal{P}_{isg} = \{p \in \mathcal{P}_i \mid r_g < p \leq p_s^+\}$ is the set of pull-outs between the arrival of g and the first pull-out p_s^+ after time s . Given c_{is} we can then form the objective by multiplying each c_{is} with the variable x_{is} , as described in the following section.

3.3 An integer programming model

We are now ready to formulate the full integer programming model, including sequencing constraints on the classification tracks. We use binary variables x_{is} that indicate at which time s the reservation of a classification track for outbound train i starts, as well as binary variables $y_{i\theta}$ that indicate whether the outbound train i is allocated to track θ .

$$\text{minimize} \quad \sum_{i=1}^n \sum_{s \in \mathcal{S}_i} c_{is} x_{is} \quad (1)$$

$$\text{subject to} \quad \sum_{\theta \in L_i} y_{i\theta} = 1, \quad 1 \leq i \leq n \quad (2)$$

$$\sum_{s \in \mathcal{S}_i} x_{is} = 1, \quad 1 \leq i \leq n \quad (3)$$

$$\sum_{s \in \mathcal{S}_j: s < o_i} x_{js} + y_{i\theta} + y_{j\theta} \leq 2, \quad (i, j) \in IJ, \theta \in L_i \cap L_j \quad (4)$$

$$\sum_{(i,s) \in \mathcal{A}_p} \sum_{g \in \mathcal{X}_{is}: r_g < p} \ell_g x_{is} \leq \ell^{\text{mix}}, \quad 1 \leq p \leq m \quad (5)$$

Equation (1) gives the objective in terms of the number c_{is} of extra roll-ins due to mixing, which results from using start time s for train i . Equation (2) ensures that all trains i are allocated to a track from the set L_i of feasible (w.r.t. length) tracks for i , and Equation (3) ensures that each train i has a start time s from its set of valid start-times \mathcal{S}_i . Equation (4) states that for each pair of trains (i, j) , where i leaves its classification track before j , but departs after the first group of cars of j is rolled-in, either i and j are on different tracks, or j starts its allocation of the common track θ after i has left. Formally, we define $IJ = \{(i, j) \mid 1 \leq i < j \leq n \wedge \min_{g \in \mathcal{G}_j} r_g < o_i\}$, where we assume that the outbound trains are indexed according to ascending departure times from the classification bowl. Finally, Equation (5) ensures that the mixing capacity limit is respected in each period.

In order to facilitate finding a feasible solution, we define the *feasibility mixing problem* as the mixed integer program consisting of Equations (2) to (5) with an additional continuous non-negative variable v representing “virtual mixing capacity”. The variable v is then added to the right-hand side of Equation (5). By replacing the objective Equation (1) with minimize v , one seeks to obtain mixing feasibility without specifically minimizing the number of extra roll-ins.

4 Case study

The Swedish traffic administration authority has provided us with historic data for the Hallsberg Rangerbangård hump yard in central Sweden for validation of our approach. Hallsberg has 8 tracks of length 595 to 693 meters on the arrival yard, two parallel humps (of which only one is in use), 32 available classification tracks of varying length (between 374 and 760 meters), and 12 tracks with length 562 to 886 meters on the departure yard. Although there are several other tracks on the yard (most notably tracks going to light and heavy repair facilities) they are not normally used for shunting, and we are therefore not considering them in the model.

We used the following working process to determine a shunting plan: First, preprocess the data according to Section 4.1. Second, determine suitable roll-in (r_i), pull-out (p_j) and roll-out times (o_j) for all inbound trains i and outbound trains j by the method described in Section 4.2. Third, separate the problem data into instances containing all rolled-in trains in a single week, ranging from Saturday until the next Friday, using the roll-in and pull-out times obtained in the second step. Fourth, solve the mixing problem instances using the heuristic algorithm in Section 2.1 and the improvement algorithm in Section 2.2. Fifth, also solve the feasibility mixing problem instances using the MIP model in Section 3.3, minimizing the virtual mixing capacity v . Finally, improve the mixing solutions obtained from the MIP model by fixing v at zero and solving the original integer program (1) to (5).

4.1 Preprocessing traffic data

We collected five months of historic traffic data for the yard, including all inbound trains, outbound trains, and the set of cars going from each inbound train to each outbound train. The data was taken from the period from December 11, 2010 to May 10, 2011, which contains intervals of both high and low activity, such as a longer holiday period and at least one major traffic disruption. Timing parameters, such as setup times, durations of roll-ins, etc., were chosen according to [1]. Cars with a local source or destination were not included in the data set and were therefore not considered.

The data we collected contain for every inbound train only the time when it arrives to the arrival yard. The time when the cars of the inbound train roll-in to the classification bowl are not available. Further, the data did not contain the time points when the mixed tracks were pulled-out. Therefore, we had to compute a hump schedule for all roll-in and pull-out operations to complete the input for the mixing problem of the previous section. This will be described in Section 4.2.

The data contain further ambiguity, namely the matching of cars from incoming trains to outgoing trains: for the same train, there may be several arrival and departure times on the same day. Further, the same physical car may arrive at the yard repeatedly, having the same ID in the data. Finally, data for trains arriving or departing outside the period for which the data was provided are missing. We therefore had to match cars to trains in a sensible manner. For our experiments, we required a minimum time span of 180 minutes between arrival at and departure from the shunting yard (i.e., not only the classification bowl). Further, we required that cars spend at most 48 hours on the yard. Finally, we required that trains do not exceed the length of the longest available classification track. Car records that could not be matched this way were discarded. In total, 3594 arrivals, 3654 departures and 17684 car groups were handled. Inbound trains vary in length between 12.8 and 929 meters, outbound trains between 12 and 1252 meters. For five outbound trains we had to discard some of their groups in order to stay below the maximal track length of 760 meters.

4.2 Computing the missing hump schedule

The mixing problem is given by the roll-in times of the inbound trains, by the departure times of the outbound trains from the classification bowl, and by the times of the pull-outs. The traffic data provided to us does not contain this information: only the arriving times to the arrival yard and the departure times from the departure yards are known. For the experiments, we therefore need to compute the missing data, trying to mimic the current practice in the yard.

Because the pull-backs as well as the roll-ins occupy the hump, we need to schedule the roll-ins and pull-backs such that no two such actions on the hump overlap in time. Additionally, we want to find a set of pull-outs that guarantees a feasible assignment and reservation of tracks to the outbound trains if we assume infinite capacity of the mixed tracks. We achieve this by creating a rough assignment of trains to classification tracks in a round-robin fashion (and ignoring mixing capacities), and computing a minimum set of pull-outs, such that there is one suitable pull-out for each train that needs to store cars on the mixed tracks in this round-robin schedule.

We assume that all tracks on the arrival and departure yards can accommodate all inbound and outbound trains. We determine the roll-in times and pull-out times sequentially by first considering the roll-ins, and then by inserting pull-outs at suitable time points. Due to space restrictions, we have to omit a detailed description of the preprocessing. We remark

■ **Table 1** Experimental results in maximum mixed track usage (MTU) with a limit of 1217 m, and the extra car-roll-ins needed due to mixing (ER). \bar{x} is the arithmetic mean. Infeasible solutions are shown in italics.

	<i>Instance</i>		<i>Heuristic</i>			<i>Heuristic++</i>			<i>MIP</i>		
	Trains (#)	Groups (#)	MTU (m)	ER (#)	Time (s)	MTU (m)	ER (#)	Time (s)	MTU (m)	ER (#)	Time (s)
1	188	901	572.7	158	0.1	526.4	150	0.6	377.3	169	1160.8
2	121	423	0.0	0	0.0	0.0	0	0.2	0.0	0	7.9
3	99	322	0.0	0	0.0	0.0	0	0.2	0.0	0	5.4
4	137	469	0.0	0	0.0	0.0	0	0.3	0.0	0	9.3
5	185	862	515.4	80	0.1	499.6	80	0.9	505.4	120	1159.4
6	184	924	249.5	66	0.1	249.5	64	0.5	455.7	111	1062.5
7	157	656	0.0	0	0.0	0.0	0	0.2	0.0	0	13.1
8	179	797	171.5	25	0.0	171.5	25	0.4	245.8	30	1085.4
9	173	856	262.1	56	0.1	262.1	51	0.5	228.9	46	1083.4
10	183	901	598.4	191	0.2	579.7	184	0.6	843.9	204	1156.0
11	185	873	952.7	133	0.1	952.7	133	0.6	374.4	55	1135.1
12	174	918	988.1	184	0.3	836.2	155	0.6	530.7	56	1129.3
13	188	930	286.3	76	0.1	286.3	76	0.4	338.3	107	1155.5
14	201	1100	1003.3	208	0.2	901.9	200	0.5	935.5	210	1157.7
15	194	1053	748.1	211	0.2	748.1	203	0.6	554.5	229	1119.7
16	173	907	<i>1365.6</i>	274	0.2	<i>1249.1</i>	258	0.6	295.7	84	1141.2
17	188	958	640.5	91	0.1	640.5	91	0.4	817.8	121	1145.4
18	199	1047	1063.9	379	0.4	1089.8	340	0.7	589.9	238	1118.5
19	156	801	1159.0	118	0.1	1032.9	106	0.4	428.5	64	1139.6
20	148	778	302.9	32	0.0	302.9	32	0.3	395.5	41	1144.7
21	186	973	<i>1797.9</i>	542	0.4	<i>1726.8</i>	530	0.8	918.6	351	1139.1
\bar{x}	171.3	830.9	603.7	134.5	0.1	574.1	127.5	0.5	420.8	106.5	917.6

that using this preprocessing we had to delay a small fraction of the trains in the five month period (0.8 % of the inbound trains for in total 82 minutes and 0.08 % of the outbound trains for in total 70 minutes). It should be noted, however, that arrival and departure times can be negotiated with the network provider. Further, in daily operations, the planners do not strictly follow a fixed rule to determine the roll-in order. In about 90% of the cases, trains are rolled-in in the same order as they arrive. In the remaining cases, the above rule of rolling in according to earliest deadline is followed.

4.3 Results

Table 1 shows the results obtained for the problem instances using the above approaches. The instances are seven days long and cover Saturday until the next Friday. In the table, *Heuristic* is the heuristic from Section 2.1, *Heuristic++* is the same heuristic improved using the algorithm in Section 2.2, and *MIP* is the integer programming model from Section 3, where we first solve the feasibility mixing problem as described above. The MIP computations were carried out using Python 2.6 and Gurobi 4.5 on a standard dual-core desktop computer. A time limit of 10 minutes for feasibility and an additional 10 minutes for optimality was imposed.

As can be seen in Table 1, both heuristics reach feasibility with regard to the total mixing capacity of 1217 m in most of the problem instances. The improvement heuristic from Section 2.2 lowers the mixed track usage and number of extra roll-ins compared to the

original coloring heuristic from Section 2.1 many instances. The MIP model finds feasible mixing solutions in all instances tried, in many cases with much fewer extra roll-ins compared to the heuristics. In addition, in 17 out of the 21 cases we could have managed with just one mixed track of length of 608 or 609 m instead of two. However, it should be noted that the runtime when optimizing the number of extra roll-ins was terminated after 10 minutes without finding an optimal solution. The MIP gap for the model minimizing the number of car roll-ins is almost always 100.0 %. At the same time, the run-time of the heuristics is negligible, as it is in the order of a fraction of a second. Although the experiments suggest that the MIP performs fast enough, we remark that faster algorithms would enable other applications as, e.g., online booking systems, similar to those in passenger traffic, where customers could make a reservation to book their cars on specific trains.

Worth noting is that instances 2–4 contain both holidays and major disruptions due to snowfall, which explains the lower load seen in Table 1. In addition, instance 7 contains a large derailment which occurred on a major line in northern Sweden. As a result, freight had to be transported on lines with lower capacity than normal, hence the lower volumes at Hallsberg during this period.

Acknowledgements

We are grateful to Per Danielsson for providing us with the visualization tool for the classification yard occupation. We would also like to thank Pertti Kuusisto and Patrik Fritzing at Trafikverket for providing the yard layout and details on shunt yard operation, and Stefan Huss at Green Cargo AB for verifying timing properties in the model.

References

- 1 C. Alzén. *Trafikeringsplan Hallsbergs rangerbangård*. Banverket, May 2006.
- 2 U. Blasum, M. R. Bussieck, W. Hochstättler, C. Moll, H.-H. Scheel, and T. Winter. Scheduling trams in the morning. *Mathematical Methods of Operations Research*, 49(1):137–148, March 1999.
- 3 F. Bonomo, G. Durán, and J. Marenco. Exploring the complexity boundary between coloring and list-coloring. *Annals OR*, 169(1):3–16, 2009.
- 4 S. Cornelsen and G. D. Stefano. Track assignment. *J. Discrete Algorithms*, 5(2):250–261, 2007.
- 5 E. Dahlhaus, P. Horák, M. Miller, and J. F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103(1–3):41–54, 2000.
- 6 E. Dahlhaus, F. Manne, M. Miller, and J. Ryan. Algorithms for combinatorial problems related to train marshalling. In *Proceedings of the Eleventh Australasian Workshop on Combinatorial Algorithms (AWOCA)*, pages 7–16, 2000.
- 7 G. Di Stefano and M. L. Koči. A graph theoretical approach to the shunting problem. *Electronic Notes in Theoretical Computer Science*, 92:16–33, February 2004.
- 8 M. Gatto, J. Maue, M. Mihalák, and P. Widmayer. Shunting for dummies: An introductory algorithmic survey. In *Robust and Online Large-Scale Optimization*, volume 5868 of *LNCS*, pages 310–337. Springer, 2009.
- 9 S. He, R. Song, and S. S. Chaudhry. An integrated dispatching model for rail yards operations. *Computers & OR*, 30(7):939–966, 2003.
- 10 R. Jacob, P. Márton, J. Maue, and M. Nunkesser. Multistage methods for freight train classification. *Networks*, 57(1):87–105, 2011.

- 11 M. W. Siddiquee. Investigation of sorting and train formation schemes for a railroad hump yard. In *Proceedings of the 5th International Symposium on the Theory of Traffic Flow and Transportation*, pages 377–387, 1972.
- 12 T. Winter and U. T. Zimmermann. Real-time dispatch in storage yards. *Annals of Operations Research*, 96(1-4):287–315, November 2000.